Nova-Wang at SemEval-2020 Task 12: OffensEmblert: an Ensemble of Offensive Language Classifiers

Susan Wang
Nova University IMS,
Lisbon, Portugal
susanwangds@gmail.com

Zita Marinho
Priberam Labs
Institute for Systems and Robotics, IST
zam@priberam.com

Abstract

This paper presents our contribution to the Offensive Language Classification Task (English SubTask A) of Semeval 2020. We propose different Bert models trained on several offensive language classification and profanity datasets, and combine their output predictions in an ensemble model. We experimented with different ensemble approaches, such as SVMs, Gradient boosting, AdaBoosting and Logistic Regression. We further propose an under-sampling approach of the current SOLID dataset, which removed the most uncertain partitions of the dataset, increasing the recall of the dataset. Our best model, an average ensemble of four different Bert models, achieved 11th place out of 82 participants with a macro F1 score of 0.91344 in the English SubTask A.

1 Introduction

As people turn more and more to the online world for their entertainment, social and communication needs, the anonymity offered by the platforms also draw out diverse and sometimes divisive opinions that can lead to abuse, bullying and mental stress. To ensure civility of online discussions while allowing for opposing ideas to be expressed, we need an effective way to detect offensive language in social media.

The SemEval 2020 Task 12: *"OffensEval 2: Multilingual Offensive Language Identification in Social Media"*(Zampieri et al., 2020), introduces a multi-lingual dataset with 5 languages (Arabic, Danish, Greek, Turkish and English). It is a continuation and extension of SemEval 2019 Task 6: *"OffensEval: Identifying and Categorizing Offensive Language in Social Media"*(Zampieri et al., 2019b).

This paper presents the model submitted to the English stream of SubTask A, focusing on the Offensive Language Classification, where for a given tweet we classify it as offensive (OFF) if it contains any form of profanity or targeted offense, either veiled or direct, and non-offensive (NON) otherwise.

The official OffensEval 2020 English Training Dataset (Rosenthal et al., 2020) was labelled using a semi-supervised method with several different models. The dataset provides a score which corresponds to an average prediction confidence of being offensive over all the models (μ) along with the uncertainty of the prediction given by their standard deviation (σ).

The main challenge of using this dataset concerns with how to make an effective use of this large dataset as the tweets with mid-range average confidence values (μ) often have high variance (σ) as a result are imprecise and difficult to interpret. To overcome this challenge, we created two sub-sampled datasets (A & B), by removing majority of mid-range score μ (uncertain predictions) and with high variance σ from both sets, and by evenly sample from positive and negative values on the second set (set B).

We also incorporated additional datasets into our model to balance out the uncertainty of the semi-supervised set. Rather than creating one single combined dataset, we trained separate models for each different dataset and experimented with various ensemble techniques. This allowed for more flexibility in tuning the impact of each dataset and allowed us to get the benefit of larger datasets without undermining smaller datasets.

We fine-tuned a BERT model for each dataset. We chose this model architecture as it has been shown to outperform models built using other structures for offensive language tasks.(Nikolov and Radivchev, 2019;

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1587–1597
Barcelona, Spain (Online), December 12, 2020.

Class	OLID Train	SOLID	Kaggle	Profanity	SOLID A	SOLID B	OLID Test	SOLID Test
NON	8,840 (67%)	7,628,650 (84%)	1,660,540 (92%)	147,509 (80%)	5,636,935 (87%)	515,000 (50%)	620 (72%)	2,807 (72%)
OFF	4,400 (33%)	1,446,768 (16%)	144,334 (8%)	36,845 (20%)	827,353 (13%)	515,000 (50%)	240 (28%)	1,080 (28%)
TOTAL	13,240 (100%)	9,075,418 (100%)	1,804,874 (100%)	184,354 (100%)	6,464,288 (100%)	1,030,000 (100%)	860 (100%)	3,887 (100%)

Table 1: Target Distribution of the datasets.

Uglow et al., 2019) We submitted an average weighted ensemble, combining output of four individual models using equal weights. It improved upon the results of individuals by generally reducing the average number of false positive and false negative results. Our model ranked 11th out of 82 on the English SubTask A[1].

2 Related Work

The abusive and offensive language classification task has been studied from different perspectives, ranging from aggression identification (Kumar et al., 2018; Modha et al., 2018), bullying detection (Xu et al., 2012), hate speech identification (Davidson et al., 2017), toxic comment classification (Fortuna et al., 2018), and offensive language classification (Wiegand et al., 2018; Zampieri et al., 2019b).

Last year edition of OffensEval 2019 proposed a three-level hierarchical schema for offensive language classification (Zampieri et al., 2019a; Zampieri et al., 2019b). This dataset considers whether (i) a tweet is offensive or not, (ii) whether the offense is targeted, and (iii) whether it is targeted towards individuals, groups or others. The associated tasks were divided into three corresponding sub-tasks. For the first sub-task, offensive language classification, models with BERT (Devlin et al., 2019) architecture consistently out-performed other methods and were used by seven of the top ten teams. We followed a similar method to the top teams, Liu et al. (2019) and Nikolov and Radivchev (2019), using their recommended steps for pre-processing the tweets and fine-tuning pre-trained BERT model.

In a related Kaggle competition in 2019, Jigsaw published a dataset for identifying toxicity and minimising bias in online comments (Jigsaw, 2019). The winning model was a blend of 2x XLNet, 2x BERT and GPT2 medium (Prokoptsev et al., 2019). We did not follow the approach due to time and resource constraints and instead opted to use the dataset to supplement our existing data.

Waseem et al. (2017) proposed a separate two-fold topology for synthesizing different subtasks in abusive language detection by considering whether (i) the abuse is directed at a specific target and (ii) the degree to which it is explicit. They noted that implicit abuse is more difficult to identify, sometimes requiring more detailed annotation guidelines and perhaps even expert annotators.

3 OffensEval proposed system

We trained several models using different datasets and combined the best ones in an ensemble. In this section we describe the datasets that were considered and how the models were trained. Table 1 outlines the target distribution of each of the datasets.

3.1 Semi-Supervised Dataset for Offensive Language Identification (SOLID)

The official OffensEval 2020 English Training Dataset (**SOLID**) (Rosenthal et al., 2020) contains over nine million tweets and was labelled in a semi-supervised manner using model built from an ensemble of PMI (Turney and Littman, 2003), LSTM (Hochreiter and Schmidhuber, 1997), FastText (Joulin et al., 2016) and BERT (Devlin et al., 2019). It followed the annotation guideline as OLID (Zampieri et al., 2019a), where a tweet is labeled as offensive (OFF) if it contains any form of profanity or targeted offense, and non-offensive (NON) otherwise. However, instead of the binary label, two numerical scores are provide for each tweet – μ and σ. μ represents the average of the confidences predicted by the models for belonging to the positive class (OFF). σ is the confidences' standard deviation.

In order to find a suitable decision boundary, we binned μ and σ and analysed their respective distribution using histograms for the confidence levels (Figure 1-middle) and standard deviation (Figure 1-bottom), as

[1]as team *M20170548* in (Zampieri et al., 2020)

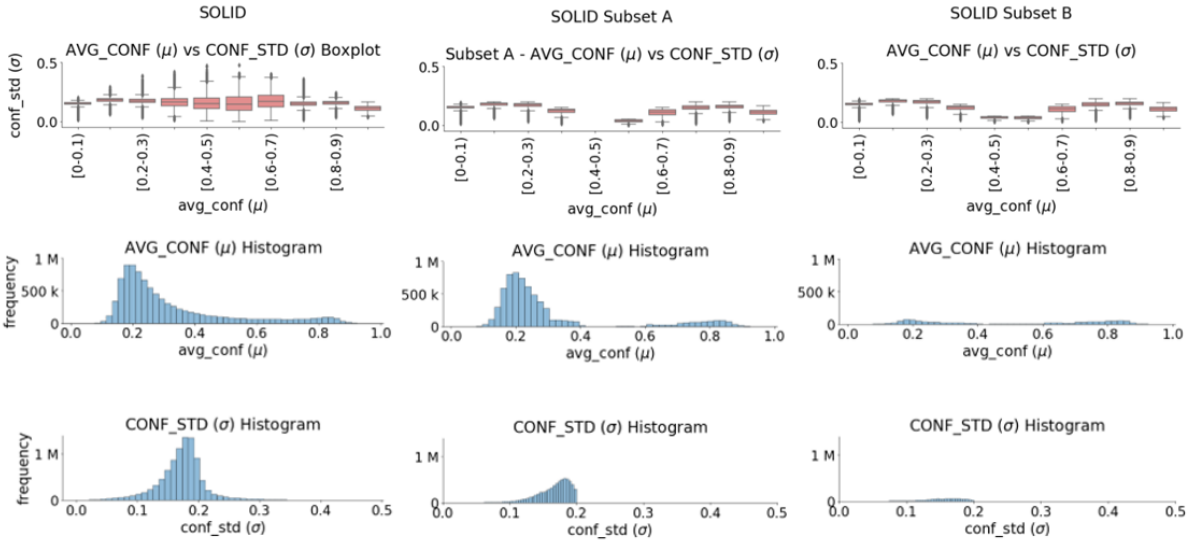

Figure 1: SOLID Data Distributions - Original (left), Subset A (centre) and Subset B (right)

well as the distribution of the standard deviation over the confidence scores (Figure 1-top). The left-most column on Figure 1 shows the dataset has significantly more negative examples (NON) than positive (OFF), and that mid-range values of μ have a higher standard deviation σ.

We inspected the tweets with mid-range values of μ ($0.4 < \mu < 0.6$) and high standard deviation σ (> 0.3) and found various examples of misclassification, assuming that the classification threshold was set at $\mu = 0.5$. The most prominent being that many tweets containing profanity had $\mu < 0.5$, as shown in Table 4 of Appendix. Additionally, self deprecating comments were often mislabelled as being offensive and general negative comments on society or environment were also incorrectly marked as offensive, see Table 5 of Appendix. As tweets with mid-range μ and / or high σ appear to be misleading, we hypothesised that training with datasets without these uncertain cases would improve the results. We created two subsets by under-sampling the mid-range values of μ ($0.4 < \mu < 0.6$) to see if a more selective sampling would improve the results, one containing about 6M samples by just under-sampling the uncertain region (SOLID A) and another with only 1M samples (SOLID B).

SOLID A was created by removing majority of records with mid-range μ as well as removing records with $\sigma > 0.2$. The distribution is shown in middle column of Figure 1. The resulting dataset contains 6,464,288 records, roughly 72% of the original dataset volume.

SOLID B was created with the intention of balancing dataset by sampling equal number records from both high and low μ ranges. Records with $\sigma > 0.2$ were removed and only a small number of tweets with mid-range μ were kept. The distribution is shown in the right column of Figure 1. The resulting dataset contains 1,030,000 records, roughly 11% of the original dataset volume.

3.2 Offensive Language Identifcation Dataset (OLID)

The Offensive Language Identification Dataset (OLID)[2] was created for the OffensEval 2019 shared task. The training set contains 14,100 manually annotated tweets, where a tweet was labelled as offensive (OFF) if it contains any form of profanity or targeted offense, either veiled or direct, and non-offensive (NON) otherwise. The ratio of OFF to NON is roughly 1 to 2. The quality of this dataset is better and more reliable than SOLID, however it is almost 650 times smaller.

3.3 Jigsaw Unintended Bias in Toxicity Classification Dataset (Kaggle)

Th Kaggle 2019 Toxicity Classification Dataset (Kaggle)[3] dataset contains over 1.8 million public comments from online news discussions. This dataset was created with the aim of reducing unintended

[2] https://scholar.harvard.edu/malmasi/olid

[3] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

bias in toxicity classification as a result of identity mentions. The data has been labelled with identity mentions, such as Muslim, Gay or Black, and a toxicity score (TARGET) that represents the faction of human annotators who believe the post is toxic. We decided to include this as it is a large dataset where each comment has been reviewed by up to 10 annotators, and its content could prove useful in reducing false positive errors.

3.4 Profanity Check (Profanity)

A simple text search of the word "fuck" in SOLID returned 268,845 matches, roughly 3% of all tweets. Out of these, 2.3% were misclassified as non-offensive. Figure 3 of the Appendix compares the distribution of the confidence scores μ and σ for profanity tweets containing the word "fuck" against those in the whole dataset. Most of the uncertainty of the predictions of profanity tweets coincided with the misclassification (Figure 3, top-left). This suggests a need to increase our model's sensitivity to profanity words. Rather than using a dictionary based approach (Han et al., 2019), we decided to use a Python library, *Profanity-check*[4], that checks for profanity and offensive language in text. It was built using a SVM model trained on 184,354 records from a twitter dataset[5] and a Wikipedia dataset (Zhou, 2019) [6].

3.5 Preprocessing

For Twitter datasets OLID (§ 3.2) and SOLID (§ 3.1) we used a Python library, emoji[7] to replace emojis with text descriptions. Ampersands (&) were replaced with "and". We expanded hashtags and contractions and removed all symbols except for the following [-?.,!@#]. For the Kaggle dataset (§ 3.3), we simply expanded the contractions. For instance, "aren't" was expanded to "are not".

3.6 Ensemble Model

We used the HuggingFace's implementation of pre-trained BERT *base-uncased* model (Wolf et al., 2019) as the basis for each model, and fine-tuned the results for each dataset independently. The learning rate was set at $lr = 2e^{-5}$, with Adam Optimizer and 5 % of the samples were used for warm up with linear schedule and batch size of 32. The classifier thresholds were set to 0.5 for all models. Starting from the pre-trained model, we fine-tuned separate models using the datasets described above (SOLID, OLID, Kaggle, Profanity) and fine tuned the results over 2-3 epochs. We combined all four models into a ensemble trained with different approaches, a simple average, using gradient boosting (Mason et al., 1999), AdaBoost (Freund and Schapire, 1995), SVMs (Vapnik, 1995) with a linear and Radial Basis Function (RBF) kernel, and logistic regression. In Table 9 in the Appendix we show the values of hyper-parameter tuning done for each model.

	Dev Set				Test Set			
Model	macro-F1	P	R	acc.	macro-F1	P	R	acc.
Profanity	0.68617	**0.85194**	0.66230	0.80581	0.85402	0.89636	0.82852	0.89220
Kaggle	0.74961	0.84081	0.72009	0.83023	0.90998	**0.90436**	0.91619	**0.92668**
SOLID A	0.78802	0.77548	0.81384	0.80833	0.90822	0.88877	0.94439	0.92050
SOLID B	0.80237	0.79466	0.81223	0.83605	0.90814	0.88877	0.94382	0.92050
SOLID	0.80967	0.84332	0.78925	0.85814	0.91136	0.89223	0.94510	0.92359
OLID	0.81122	0.80278	**0.82218**	0.84302	0.90428	0.88543	0.93798	0.91742
Avg Ensemble	**0.81219**	0.84795	0.79086	**0.86047**	**0.91344**	0.89464	**0.94539**	0.92560

Table 2: Results of individual models, reported macro-F1, precision, recall and accuracy for each model. We show the best model in bold

[4] https://pypi.org/project/profanity-check/

[5] https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data

[6] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[7] https://pypi.org/project/emoji/

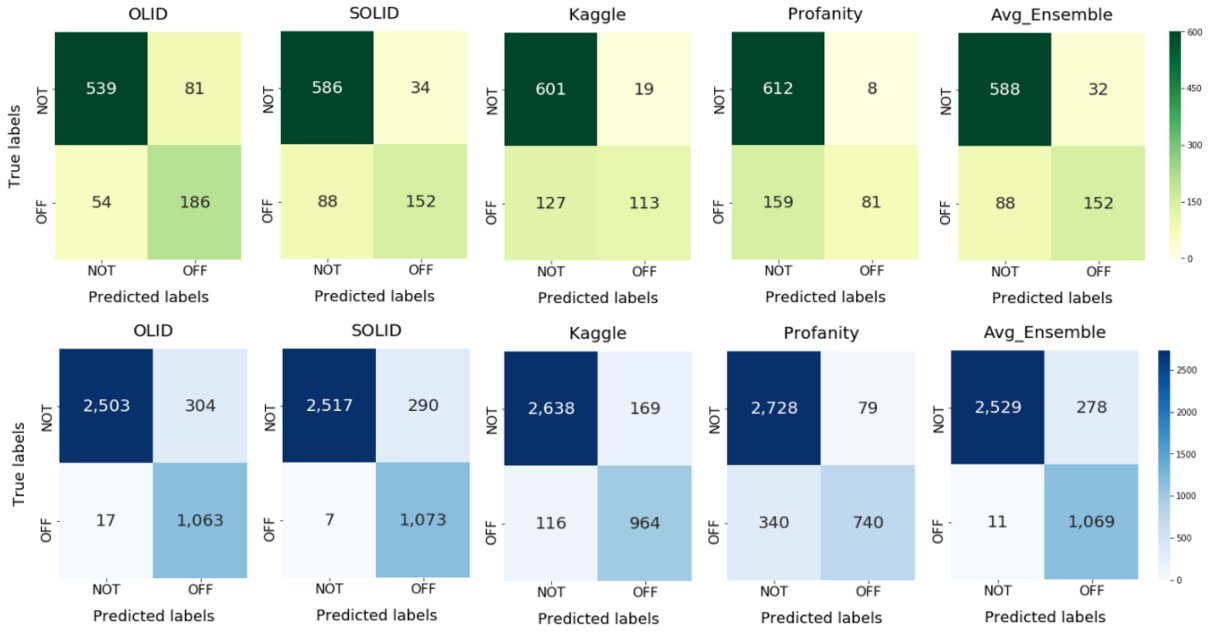

Figure 2: Confusion Matrices on the dev set (top), and test set (bottom), showing each model has different inductive bias.

4 Experimental Results

We created train/dev splits with the sizes specified in Table 8 for hyper-parameter tuning for learning each model individually. We further used OLID test set as the validation set for model evaluation and selection.

We report the official metric for this task macro-F1, giving equal importance to precision and recall, as well as equal weighting on the minor and major classes. Due to the imbalanced data, the performance of the model on minority class was particularly important.

Table 2 details the classification results of each model independently. We report the validation scores (OLID Test) and the test scores (SOLID Test). In addition to macro-F1, we also also included macro precision (P), recall (R) and accuracy (acc.). OLID and SOLID models show better recall, while Kaggle and Profanity models had better precision. This is also reflected in the confusion matrices on Figure 2, where OLID and SOLID had significantly less false negatives while Kaggle and Profanity have far less false positives.

Contrary to our initial hypothesis, the models trained from under-sampling the uncertain partition of SOLID did not perform as well as the model built using the whole set. We posit that further tuning of the optimal threshold μ and σ for under-sampling, may influence the results. Possibly apply a more aggressive under-sampling could increase precision scores. Due to time constraints we leave further exploration as future work.

Seeing that each model has different strengths, we further experiment with various ensemble techniques and combine these four models - OLID, SOLID, Kaggle and Profanity.

We reported the best model on the dev set (OLID Test). The average weighted ensemble (Avg.) used equal weight for each of the four component models. For the grid search ensemble (Grid), we searched weights from 0 to 1, in steps of 0.1. This resulted in over 1000 combinations with the same best results, possibly over-fitting due to the small dev set size ($\tilde{8}60$ examples). For the remainder of the ensembles, we used 10 fold cross validation combined with grid search on the hyper-parameters to find the best parameters for the ensembles. Table 9 in the Appendix summarizes the hyper-parameters we explored.

We report the model ensemble results in Table 3.[8] We submitted the Avg ensemble, since this was the

[8]We applied the Wilcoxon Rank-Sum Test to compare the prediction distributions of each ensemble against the predictions by the average weighted ensemble for the dev set, and found that only the grid search ensemble produced a result that is statistically different ($p<0.01$).

	Dev Set				Test Set			
Models	**F1**	**P**	**R**	**acc.**	**F1**	**P**	**R**	**acc.**
Logistic Regression	0.80784	0.84698	0.78542	0.85814	0.91212	0.89358	0.94325	0.92462
AdaBoost	0.80848	0.85073	0.78495	0.85930	0.91304	0.89442	0.94436	0.92539
SVM linear	0.80967	0.84332	0.78925	0.85814	0.91136	0.89223	0.94510	0.92359
Avg	0.81219	0.84795	0.79086	0.86047	**0.91344**	**0.89464**	**0.94539**	**0.92565**
Gradient Boost	0.81414	**0.85400**	0.79120	**0.86279**	0.91244	0.89385	0.94371	0.92488
SVM RBF	0.81474	0.85269	0.79247	**0.86279**	0.91076	0.89208	0.94265	0.92333
Grid	**0.81528**	0.80540	**0.82890**	0.84535	0.90521	0.88597	0.94090	0.91700

Table 3: Results of ensembles, reported macro F1, precision (P), recall (R) and accuracy (acc.) for dev and test sets. Bold values show the best performant models

only we had experimented at the time of submission. This model was also the best ensemble in the test set.

5 Conclusions and Future Work

There are offensive language classification datasets that portray different annotation guidelines and different annotators' subjectivity. In this world we take advantage of the heterogeneous nature of these datasets to improve the performance of offensive language identification, by combining several models in an ensemble. This showed the importance of having training data that is reliable and diverse enough to capture different types of scenarios, and the potential benefit of combining and consolidating those datasets. We hypothesised that under-sampling uncertain and possibly mis-classified tweets we could improve the performance of classification algorithms, but the results so far have proven to be inconclusive. We leave as future work more aggressive under-sampling schemes to access the consequences of only training on highly confident predictions, and differ additional semi-supervised strategies to improve results using SOLID dataset. We show that using different sources for training offensive language classification tasks helps improve the quality of the predictions.

References

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Paula Fortuna, José Ferreira, Luiz Pires, Guilherme Routar, and Sérgio Nunes. 2018. Merging datasets for aggressive text identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 128–139, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Yoav Freund and Robert E. Schapire. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In Paul Vitányi, editor, *Computational Learning Theory*, pages 23–37, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jiahui Han, Shengtan Wu, and Xinyu Liu. 2019. jhan014 at SemEval-2019 task 6: Identifying and categorizing offensive language in social media. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 652–656, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Conversation AI Jigsaw. 2019. Civil comments annotated with toxicity and identity content kaggle competition. *https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

L. Mason, J. Baxter, P. L. Bartlett, and Marcus Frean. 1999. Boosting algorithms as gradient descent in function space. *http://sciencewise.info/resource/Gradient$_b$oosting/Gradient$_b$oosting$_b$y$_W$ikipedia.*

Sandip Modha, Prasenjit Majumder, and Thomas Mandl. 2018. Filtering aggression from the multilingual social media feed. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 199–207, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Alex Nikolov and Victor Radivchev. 2019. Nikolov-radivchev at semeval-2019 task 6: Offensive tweet classification with bert and ensembles. In *SemEval@NAACL-HLT*.

Nikolay Prokoptsev, Gennady Shtekh, and Yury Bolkonskiy. 2019. 1st place solution in jigsaw unintended bias in toxicity classification. *https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/discussion/103280*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *ArXiv*, abs/2004.14454.

Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *CoRR*, cs.CL/0309034.

Harrison Uglow, Martin Zlocha, and Szymon Zmyslony. 2019. An exploration of state-of-the-art methods for offensive language detection. *CoRR*, abs/1903.07445.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. *CoRR*, abs/1705.09899.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. 09.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 656–666, Montréal, Canada, June. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

Victor Zhou. 2019. Building a better profanity detection library with scikit-learn. *https://victorzhou.com/blog/better-profanity-detection-with-scikit-learn/*.

Appendix

A SOLID examples

Here are some examples of mis-classified tweets from the semi-supervised training set. Table 4 shows many tweets containing profanity had $\mu < 0.5$. Table 5 shows non-offensive tweets being mis-classified with $\mu > 0.5$. Often the latter contain self-deprecating comments or negative statements or emotions that are not in fact offensive.

text	average	std
First of all bitchLet me take Lay to Walmart before I push her down	0.400922	0.339073
Not in da mood 4 these hoes	0.401108	0.321429
my favorite part of coming to the gym alone is that all the persian fuckboys come out to play	0.420394	0.321399
But America is the shittiest place on earth right?? ??	0.423781	0.319078
I do love all the slutty costumes though. Even though half of you are sluts anyways.	0.424623	0.314600
religion is so fuckingggff bulshittt	0.426189	0.300384
Thick master Ken is certainly one of the nastiest sissy boys of all time	0.436352	0.303370
So Lion King was absolute dogshit	0.442239	0.315102
@USER Tell it to the IMBECILE in the White House! His name is #Trump	0.443468	0.314032
@USER You are Corrupted! You should retire right now and step down!	0.480829	0.313158
Sooo many fresh when we was in school niggas bums now....tf happen to u!	0.485208	0.318131
"@USER only If you are weak and insecure"	0.489890	0.308845
The Chinese Satanist	0.490647	0.327647
The next President shouldn't be personally vile.	0.492065	0.302249
"@USER attention WHORE be gone."	0.493955	0.320506
Losing to Retards	0.496200	0.303528

Table 4: Example of offensive texts with $\mu < 0.5$ in SOLID

text	average	std
Today sucked. Where's the tequila?	0.501158	0.321304
caring about something you know is pointless	0.503435	0.311307
fear and greed	0.504076	0.305328
This heat is truly obnoxious	0.504384	0.303848
My body is garbage and I would love to rent a new one pls.	0.523693	0.325015
Let me start on this book to kill some time	0.536988	0.323009
You know for a bimbo I sure do read a lot...	0.538729	0.343389
in a world that needs a lot of help i feel so useless ...	0.545438	0.302742
Ate veggies today and I'm not thinner...damn it!	0.547136	0.314980
Animals is one hell of an album	0.547262	0.312807
The world is cruel #bot	0.547752	0.317849
@USER It's coming. Be patient	0.548401	0.325815
Ignorance is a bliss	0.548738	0.300488
the graveyard 2 was insane	0.549510	0.331272
@USER i will, thank you. this week was crap.	0.550122	0.319092
Still missing you like crazy tho'	0.557427	0.338659
I am just too sad right now and i feel worthless...	0.557784	0.347495
I've really, really been the best of fools, I did what I could	0.561318	0.300886
Why do fools fall in love?	0.575715	0.330991

Table 5: Example of non-offensive texts with $\mu > 0.5$ in SOLID

B Distribution of examples containing f**k word

Figure 2 compares the distribution of the confidence scores (μ) and (σ) for profanity tweets containing the word "fuck" against those in the whole dataset. The box-plot on top left show that tweets with μ between 0.30.4 have a larger σ, indicating more uncertainty in the classification. The SOLID-AVG_CONF(μ) histogram in the middle left also shows some tweets with the word "f**k" had μ 0.5

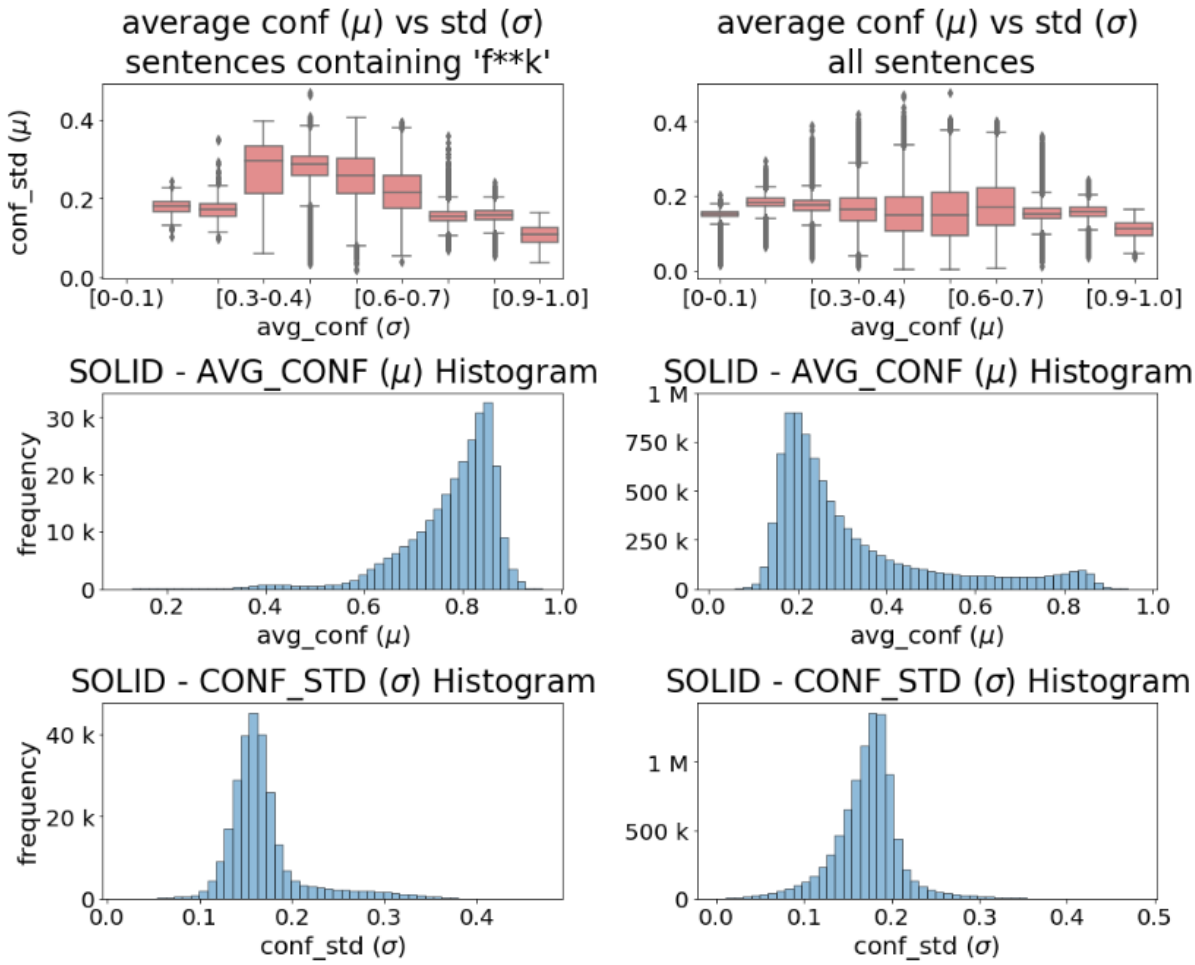

Figure 3: Distribution of f**k words in SOLID

C Classification examples for each model

tweet	actual_label	olid_pred	solid_pred	kaggle_pred	prof_pred	avg_ens_pred
And that's why you were at NYFW cause your such a class act.. getcho wack ass outta here URL	1	1	1	1	1	1
@USER @USER @USER @USER Then why aren't there so many shootings in Virginia? Jacksonville shooter got his guns IN MARYLAND!! Among the poster children for gun control! What good did that do?! You're full of crap.	1	1	1	1	0	1
#Kids Beating #Paraplegic #Trump #Boy! #Left Teach Their #Children Hate Violence! #schools #teachers #Democrats #Liberals #WalkAway #Moms #Mother #NEA #teachthemyoung #Christian #Catholic #Jews #Evangelicals #family #BlackWomen #BlackTwitter #Blacks #News #Media #Disabled #MAGA URL	1	0	0	1	1	1
#EmmyAwards2018 - Ratings tank as expected. Who can possibly stand to watch so many liberals in one place at one time.	1	0	0	0	0	0
#DregsOfSociety ??? #Deplorables Liberals usually dehumanize those they want to exterminate later URL	1	1	1	0	0	1
#Liberals Are Reaching Peak Desperation To Call On #PhillipRuddock To Talk With #Turnbull To Convince Him To Help with #WentworthVotes 18 Sept 2018 @USER #Auspol #LNP #NSWpol @USER @USER @USER #LNPMemes URL	1	0	0	0	0	0
#OrrinHatch I can't believe this sexist , clueless, old fart gets to weigh in on another woman's charges against a Supreme Court nominee. And he is spouting the same old nasty shit he spewed 20+ years ago against Anita Hill. His time's up! Good riddance Neanderthal!	1	1	1	1	1	1
The Dregs of Society"... more LABELINGand DEHUMANIZING Conservatives.I suggest,WE, Conservatives keep OUR ammo dry and OUR guns loaded..If the Fascist Liberal Left THINK Conservatives WILL go quietly into internment camps or slaughtered in the streets they NEED a REALITY check. URL	1	1	1	0	0	1

Table 6: Dev Set (OLID Test) - predictions by models when the actual label is "OFF"

tweet	actual_label	olid_pred	solid_pred	kaggle_pred	prof_pred	avg_ens_pred
#Antifa demo in memory of #KillahP. #Greece URL	0	0	0	0	0	0
#BBCNews - Cleverly: There have been no cuts to the #NHS James Cleverly on #Conservatives and coalition health spending URL	0	0	0	0	0	0
3 days before BBC Radio 2's Festival in a Day, they decide to tell us that their headline act isn't coming anymore. Thanks a lot, now I've paid a fortune to see a tonne of acts who are shit and who are of no interest to me	0	1	1	1	0	1
#gmm #roastmycat hi @USER this is Fiona. She is about 10 weeks old. She falls asleep in weird places URL	0	0	0	0	0	0
#Jenelle wants the world to know she is in a bikini. Oh, and to pray for NC. URL	0	0	0	0	0	0
#Kavanaugh so a wild claim from 36 years ago of groping has evolved into A Rape A Violent Sexual Event by Move URL a Soros based Org. that supports BLM Antifa etc. Unbeievable!	0	1	1	0	0	1
#TRUMP: I'd be a real good witness! #JOHNDowd: "You are #not a good witness. Mr. President, I'm afraid I just can't help you!" The day following those remarks, Attorney Dowd quit! EVERYBODY... URL	0	0	0	0	0	0
#JusticeForSoniasFather @USER @USER @USER @USER @USER @USER @USER @USER Plz help Sonia. She is in pain. Why PTI officials not responding her. She is also getting trouble from PMLN goons. Some1, plz help her. Thanks	0	1	0	0	0	0

Table 7: Dev Set (OLID Test) - predictions by models when the actual label is "NON"

D Individual model train/dev sizes

Model	Training Set Size	Validation Set Size
OLID	11,916	1,324
SOLID	9,075,365	50,000
SOLID subsample A	6,444,288	20,000
SOLID subsample B	1,010,000	20,000
Kaggle	1,624,387	180,487
Profanity Check	N/A	N/A

Table 8: Training and Validation Set Sizes.

E Ensemble hyper-parameter tuning

Here we show the hyper-parameter search for each ensemble model, where
. n_estimators = The maximum number of estimators at which boosting is terminated
. C = Inverse of regularization strength; smaller values specify stronger regularization
. gamma = inverse of the radius of influence of samples selected by the model as support vectors

Ensemble Method	Parameter Search	Parameter Chosen
average weighted ensemble	N/A	equal weights of 0.25
grid ensemble	check weights between 0 and 1 in incremental steps of 0.1	w(olid) = 0.5, w(solid) = 0.3, w(kaggle) = 0.5, w(prof) = 0.3
gradient boosting	n_estimators: [5, 10, 30, 50]	n_estimators = 10
ada boost	n_estimators: [5, 10, 30, 50]	n_estimators = 10
svm - rbf kernel	C : [0.1, 1, 10, 100] gamma: [0.01, 0.1, 1, 10]	C = 100 gamma = 0.1
svm - linear kernel	C : [0.1, 1, 10, 100]	C = 0.1
logistic regression	C : [0.1, 1, 10, 100]	C = 0.1

Table 9: Parameters of Ensembles